

# METHOD AND APPARATUS FOR BUILDING COMMERCIAL DISTRIBUTED COMPUTING NETWORKS VIA COMPUTER COST SUBSIDIZATION

## BACKGROUND OF THE INVENTION

5

1. Technical Field:

The present invention relates generally to an improved data processing system, and in particular to a method and apparatus for processing data. Still more particularly, the present invention provides a method, apparatus, and computer implemented instructions for distributed computation using subsidized computer systems.

15 2. Description of Related Art:

The Internet, also referred to as an "internetwork", is a set of computer networks, possibly dissimilar, joined together by means of gateways that handle data transfer and the conversion of messages from a protocol of the sending network to a protocol used by the receiving network. When capitalized, the term "Internet" refers to the collection of networks and gateways that use the TCP/IP suite of protocols.

The Internet has become a cultural fixture as a source of both information and entertainment. Many businesses are creating Internet sites as an integral part of their marketing efforts, informing consumers of the products or services offered by the business or providing other information seeking to engender brand loyalty.

Additionally, many federal, state, and local government agencies are also employing Internet sites for informational purposes, particularly agencies, which must

interact with virtually all segments of society such as the Internal Revenue Service and secretaries of state. Providing informational guides and/or searchable databases of online public records may reduce operating costs.

5 Further, the Internet is becoming increasingly popular as a medium for commercial transactions. With respect to commerce on the Web, individual consumers and business use the Web to purchase various goods and services. In offering goods and services, some companies offer goods  
10 and services solely on the Web while others use the Web to extend their reach.

Another benefit made possible by the Internet is to enable processing of problems, which are intractable using modern day super computers. With the Internet, it is  
15 possible to take many of these "intractable" problems, divide them into small work units, and have each work unit process by one of hundreds, thousands, or even millions of network-connected computers.

Many large computation tasks can be decomposed into  
20 a number (sometimes a large number) of smaller computation tasks which, when their results are properly integrated, provide the solution to the overall task. SETI@home is a popular example of one such large task: a large amount of radio telescope data is collected, and  
25 small snippets of the data are sent out to subscribing machines around the world. The data is processed, the results are returned to the central collection point, and eventually all of the data is processed. Other tasks that are easily decomposed and distributed in this manner  
30 are Monte Carlo simulations and computer graphics (for instance, ray tracing of complex scenes). A work unit is typically a collection of data and functions that operate

on that data. For instance, using an Object-Oriented programming language such as Java, a work unit might be a Class or a set of Classes that contain a set of data, plus methods to process that data. When a work unit is completed at a computer, the computer returns the result. Then, the computer may receive another work unit for processing. The collection of computers providing this distributed computing function also is referred to as a distributed computing array.

10 In many cases, the requestor or customer for which the work units are being processed is charged for use of the distributed computing array. The ability to use a distributed computing array for a charge is desirable to a customer because the customer does not have to build or 15 maintain a distributed computing array. Oftentimes, the customer may only have an occasional need for this type of computing power.

20 On the other hand, the provider of the distributed computing array is left with the burden of building and maintaining the distributed computing array. For example, real estate or space is required for housing the computers or work stations. Additionally, maintenance and power costs are present. The time and expense involved with building and maintaining the distributed computing array 25 reduces the number of businesses or organizations that are able to provide these types of services.

30 In some cases, a business may obtain the use of computing time from owners of individual computers. In many cases, the use of computing time comes at a cost in which the owners request or require payment be made for the use. The management of this type of distributed computing array also may be prohibitive. For example, the

work units processed are tracked for each computer owner and a check must be generated and sent to each individual. This may result in a requirement to track and send checks to thousands of individuals.

5        It would be advantageous to have a method and apparatus for providing a large distributed computing array to process the work unit elements of these large problems.

44-32000-10288US1

## SUMMARY OF THE INVENTION

5 The present invention provides for a method and apparatus for distributed computing in a network data processing system. A task is accepted for distributed computing. Work units are sent to a plurality of data processing systems on a network, wherein each data  
10 processing system within the plurality of data processing systems includes a software for accepting a work unit, processing the work unit to generate a result, and returning the result, wherein the software is monitored for compliance with an operation policy requiring a  
15 connection to the network and allocating a period of time for processing work units. Results are received from the plurality of data processing systems. These data processing systems may be individually owned by consumers in which the systems are provided at no cost or some  
20 lower cost in return for making available processing resources for processing work units.

**BRIEF DESCRIPTION OF THE DRAWINGS**

The novel features believed characteristic of the  
5 invention are set forth in the appended claims. The  
invention itself, however, as well as a preferred mode of  
use, further objectives and advantages thereof, will best  
be understood by reference to the following detailed  
description of an illustrative embodiment when read in  
10 conjunction with the accompanying drawings, wherein:

**Figure 1** is a pictorial representation of a network  
of data processing systems in which the present invention  
may be implemented;

15 **Figure 2** is a block diagram of a data processing  
system that may be implemented as a server in accordance  
with a preferred embodiment of the present invention;

**Figure 3** is a block diagram illustrating a data  
processing system in which the present invention may be  
implemented;

20 **Figure 4** is a diagram illustrating a distributed  
computing network in accordance with a preferred  
embodiment of the present invention;

**Figure 5** is a diagram illustrating software  
components used in distributed computation in accordance  
25 with a preferred embodiment of the present invention;

**Figure 6** is a diagram of a request in accordance  
with a preferred embodiment of the present invention;

30 **Figure 7** is a flowchart of a process used for  
setting up a computer in a distributed computing array in  
accordance with a preferred embodiment of the present  
invention;

TOP SECRET//COMINT  
REF ID: A6572

**Figure 8** is a flowchart of a process used for processing a task or request received from a customer in accordance with a preferred embodiment of the present invention;

5       **Figure 9** is a flowchart of a process used for processing work units in accordance with a preferred embodiment of the present invention;

10      **Figure 10** is a flowchart of a process used for monitoring work units in accordance with a preferred embodiment of the present invention;

15      **Figure 11** is a flowchart of a process used for enforcing a policy for running software to process work units in accordance with a preferred embodiment of the present invention; and

20      **Figure 12** is a flowchart of a process used for enforcing a policy for processing work units in accordance with a preferred embodiment of the present invention.

**DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT**

5       With reference now to the figures, **Figure 1** depicts a pictorial representation of a network of data processing systems in which the present invention may be implemented. Network data processing system **100** is a network of computers in which the present invention may be  
10 implemented. Network data processing system **100** contains a network **102**, which is the medium used to provide communications links between various devices and computers connected together within network data processing system **100**. Network **102** may include connections, such as wire, wireless communication links, or fiber optic cables.

15       In the depicted example, server **104** is connected to network **102** along with storage unit **106**. In addition, clients **108**, **110**, and **112** are connected to network **102**. These clients **108**, **110**, and **112** may be, for example, personal computers or network computers. In the depicted example, server **104** provides data, such as boot files, operating system images, and applications to clients **108-112**. Clients **108**, **110**, and **112** are clients to server **104**. Network data processing system **100** may include additional servers, clients, and other devices not shown. In the depicted example, network data processing system **100** is the Internet with network **102** representing a worldwide collection of networks and gateways that use the TCP/IP suite of protocols to communicate with one another.  
20       At the heart of the Internet is a backbone of high-speed data communication lines between major nodes or host  
25  
30

09352354-0500

computers, consisting of thousands of commercial, government, educational and other computer systems that route data and messages. Of course, network data processing system **100** also may be implemented as a number of different types of networks, such as for example, an intranet, a local area network (LAN), or a wide area network (WAN). **Figure 1** is intended as an example, and not as an architectural limitation for the present invention.

Referring to **Figure 2**, a block diagram of a data processing system that may be implemented as a server, such as server **104** in **Figure 1**, is depicted in accordance with a preferred embodiment of the present invention. Data processing system **200** may be a symmetric multiprocessor (SMP) system including a plurality of processors **202** and **204** connected to system bus **206**.

Alternatively, a single processor system may be employed. Also connected to system bus **206** is memory controller/cache **208**, which provides an interface to local memory **209**. I/O bus bridge **210** is connected to system bus **206** and provides an interface to I/O bus **212**. Memory controller/cache **208** and I/O bus bridge **210** may be integrated as depicted.

Peripheral component interconnect (PCI) bus bridge **214** connected to I/O bus **212** provides an interface to PCI local bus **216**. A number of modems may be connected to PCI local bus **216**. Typical PCI bus implementations will support four PCI expansion slots or add-in connectors. Communications links to network computers **108-112** in **Figure 1** may be provided through modem **218** and network adapter **220** connected to PCI local bus **216** through add-in boards.

Additional PCI bus bridges **222** and **224** provide interfaces for additional PCI local buses **226** and **228**, from which additional modems or network adapters may be supported. In this manner, data processing system **200** allows connections to multiple network computers. A memory-mapped graphics adapter **230** and hard disk **232** may also be connected to I/O bus **212** as depicted, either directly or indirectly.

Those of ordinary skill in the art will appreciate that the hardware depicted in **Figure 2** may vary. For example, other peripheral devices, such as optical disk drives and the like, also may be used in addition to or in place of the hardware depicted. The depicted example is not meant to imply architectural limitations with respect to the present invention.

The data processing system depicted in **Figure 2** may be, for example, an IBM e-Server pSeries system, a product of International Business Machines Corporation in Armonk, New York, running the Advanced Interactive Executive (AIX) operating system or LINUX operating system.

With reference now to **Figure 3**, a block diagram illustrating a data processing system is depicted in which the present invention may be implemented. Data processing system **300** is an example of a client computer. Data processing system **300** employs a peripheral component interconnect (PCI) local bus architecture. Although the depicted example employs a PCI bus, other bus architectures such as Accelerated Graphics Port (AGP) and Industry Standard Architecture (ISA) may be used.

Processor **302** and main memory **304** are connected to PCI local bus **306** through PCI bridge **308**. PCI bridge **308** also

093523-54 DESIGN

may include an integrated memory controller and cache memory for processor **302**. Additional connections to PCI local bus **306** may be made through direct component interconnection or through add-in boards. In the depicted 5 example, local area network (LAN) adapter **310**, SCSI host bus adapter **312**, and expansion bus interface **314** are connected to PCI local bus **306** by direct component connection. In contrast, audio adapter **316**, graphics adapter **318**, and audio/video adapter **319** are connected to 10 PCI local bus **306** by add-in boards inserted into expansion slots. Expansion bus interface **314** provides a connection for a keyboard and mouse adapter **320**, modem **322**, and additional memory **324**. Small computer system interface (SCSI) host bus adapter **312** provides a connection for hard 15 disk drive **326**, tape drive **328**, and CD-ROM drive **330**. Typical PCI local bus implementations will support three or four PCI expansion slots or add-in connectors.

An operating system runs on processor **302** and is used to coordinate and provide control of various components 20 within data processing system **300** in **Figure 3**. The operating system may be a commercially available operating system, such as Windows 2000, which is available from Microsoft Corporation. An object oriented programming system such as Java may run in conjunction with the 25 operating system and provide calls to the operating system from Java programs or applications executing on data processing system **300**. "Java" is a trademark of Sun Microsystems, Inc. Instructions for the operating system, the object-oriented operating system, and applications or 30 programs are located on storage devices, such as hard disk drive **326**, and may be loaded into main memory **304** for

execution by processor **302**.

Those of ordinary skill in the art will appreciate that the hardware in **Figure 3** may vary depending on the implementation. Other internal hardware or peripheral devices, such as flash ROM (or equivalent nonvolatile memory) or optical disk drives and the like, may be used in addition to or in place of the hardware depicted in **Figure 3**. Also, the processes of the present invention may be applied to a multiprocessor data processing system.

As another example, data processing system **300** may be a stand-alone system configured to be bootable without relying on some type of network communication interface, whether or not data processing system **300** comprises some type of network communication interface. As a further example, data processing system **300** may be a Personal Digital Assistant (PDA) device, which is configured with ROM and/or flash ROM in order to provide nonvolatile memory for storing operating system files and/or user-generated data.

The depicted example in **Figure 3** and above-described examples are not meant to imply architectural limitations. For example, data processing system **300** also may be a notebook computer or hand held computer in addition to taking the form of a PDA. Data processing system **300** also may be a kiosk or a Web appliance.

Turning next to **Figure 4**, a diagram illustrating a distributed computing network is depicted in accordance with a preferred embodiment of the present invention. In this example, customer **400** may send a request to network array server **402** to process a task. The task may be divided into work units by customer **400** or by network

array server **402**. Upon receiving the requests, the work units may be distributed to distributed computing array **404** for processing.

As results are generated from processing work requests, the results are returned from distributed computing array **404** to network array server **402**. These results are returned to customer **400** with customer **400** being billed for the processing.

In the depicted examples, distributed computing array **404** is made up of many personal or home computers provided to users for no cost or a low cost. The cost for subsidizing these computers within distributed computing array **404** is recouped by charging customer **400** for processing the request. The computers are provided to the users under certain stipulations. For example, each computer within distributed computing array **404** includes software that accepts, processes, returns, and requests work units from the business source, which is network array server **402**, in this example. Additionally, users may be required to have the computer powered on, have network access, and run this software for some minimum number of hours a day.

Although the users bear the cost of providing network access and power for the computers, they receive a benefit from being provided a computer at a low cost or at no cost. The business setting up the array bears the burden of the initial cost of establishing the array, but maintenance of the array is reduced because accounting and reimbursing individual users for processing resources is avoided. For example, keeping track of individual work units, and generating and sending thousands of checks is avoided. Further, space and electricity needed

for the computers also is provided by the users.

Turning now to **Figure 5**, a diagram illustrating software components used in distributed computation is depicted in accordance with a preferred embodiment of the 5 present invention. In this example, server **500** receives and distributes work to data processing system **502** and data processing system **504** through network **506**. This network may be, for example, a local area network (LAN), a wide area network (WAN), an intranet, the Internet, or 10 some combination thereof. Server **500** may be implemented as network array server **402** in **Figure 4** using data processing system **200** in **Figure 2**, while data processing systems **502** and **504** may be implemented as distributed computing array **404** in **Figure 4** using data processing 15 system **300** in **Figure 3**. Of course, a distributed computing array would have hundreds or thousands of computers.

Work server **508** is a software component used to send work unit **510** to client processes **512** and **514** executing 20 on data processing systems **502** and **504**, respectively. Client processes **512** and **514** each include processes used for receiving work units, processing work units, requesting work units, and returning of results. Results returned from client processes **512** and **514** are stored in 25 results **516**.

Processing resource database **518** is accessed by work server **508** to identify and assign work units to different data processing systems. A requestor or other third party presenting the computational project may be billed 30 using billing database **520**. Billing may take various forms. For example, a customer may be billed for each

work unit, for the job in its entirety, or on a flat monthly fee.

The various components depicted in **Figures 1-5** are provided for purposes of illustration and are not

5 intended to limit the architecture or components used in implementing invention.

Turning now to **Figure 6**, a diagram of a request is depicted in accordance with a preferred embodiment of the present invention. In this example, request **600** includes

10 header **602** and body **604**. Header **602** contains client address **606** identifying the data processing system that is to process the work unit. In the depicted examples, a work unit is made up of data **608** and function **610**.

Function **610** is the functional process that is to be 15 performed upon data **608**. Priority **612** indicates a priority for the particular work unit and may be used to determine the order to which the work units are processed at a particular data processing system.

With reference now to **Figure 7**, a flowchart of a 20 process used for setting up a computer in a distributed computing array is depicted in accordance with a preferred embodiment of the present invention. The process illustrated in **Figure 7** may be implemented in a server, such as network array server **402** in **Figure 4**.

25 This process is used to receive and process requests for subsidized computers that are to be added to a distributed computing array, such as distributed computing array **404** in **Figure 4**.

The process begins by receiving a request for a 30 computer (step **700**). This request may be received as a form filled out by a user in a Web page. Next, the

building of a computer with worker software is initiated (step **702**). Step **702** may be initiated by the generation of an order for the computer within the business setting up the distributed computing array or as an order to a  
5 third party vendor. Then, the computer is shipped to the requestor (step **704**) with the process terminating thereafter.

Turning next to **Figure 8**, a flowchart of a process used for processing a task or request received from a  
10 customer is depicted in accordance with a preferred embodiment of the present invention. The process illustrated in **Figure 8** may be implemented in a server, such as server **500** in **Figure 5**.

The process begins by receiving a processing request  
15 from a customer (step **800**). The request or task may already be divided into work units or may require division by the server. A determination is then made as to whether the processing request is divisible (step **802**). If the request is already in the form of work  
20 units, then the request is not divisible. If the processing request is divisible, the processing request is divided into work units (step **804**). The database is then searched for available clients within a distributed computing array (step **806**). This search may be made by  
25 searching a database, such a processing resource database **518** in **Figure 5**. Work units are assigned to clients identified by the search (step **808**), and the work units are sent to those clients (step **810**).

Next, results are received from the identified  
30 clients (step **812**). The requestor is billed (step **814**). The results are then sent to the requestor (step **816**)

with the process terminating thereafter.

Turning back to step **802**, if the processing request is not divisible, a database is searched for a client (step **818**). In this case, the request already contains one or more work units ready for processing by a distributed computing array. The processing request is assigned to one or more identified clients (step **820**), and that request is sent to these identified clients (step **822**) with the process proceeding to step **812** as described above.

With reference now to **Figure 9**, a flowchart of a process used for processing work units is depicted in accordance with a preferred embodiment of the present invention. The process illustrated in **Figure 9** may be implemented in client process, such as client process **512** or client process **514** in **Figure 5**.

The process begins by receiving a request to process a work unit (step **900**). Next, priority is identified for the work unit (step **902**). The work unit is then processed based on priority (step **904**). A result is returned to the requestor (step **906**) with the process terminating thereafter.

Turning next to **Figure 10**, a flowchart of a process used for monitoring work units is depicted in accordance with a preferred embodiment of the present invention. The process illustrated in **Figure 10** may be implemented in client process **512** or client process **514** in **Figure 5**.

The process begins by identifying the number of unprocessed work units present in the data processing system (step **1000**). A determination is made as to whether the identified work units are less than a

threshold (step **1002**). If the number of unprocessed work units is less than a threshold, additional work units are requested for processing by the data processing system (step **1004**) with the process terminating thereafter. If

5 the number is not less than a threshold, the process returns to step **1000**.

With reference now to **Figure 11**, a flowchart of a process used for enforcing a policy for running software to process work units is depicted in accordance with a 10 preferred embodiment of the present invention. The process illustrated in **Figure 11** may be implemented in client process **512** or client process **514** in **Figure 5**.

A policy is present in this process to ensure that the computer is powered, connected to the network, and 15 running the software needed for processing work units. Failure to enforce the policy may result in the computer being repossessed. Further, software may be present to disable the computer until compliance with the policy occurs.

20 The process begins by detecting initialization of the data processing system (step **1100**). Thereafter, a determination is made as to whether the software for processing work units is detected during initialization of the data processing system (step **1102**). If the

25 software is not detected, the data processing system is disabled (step **1104**) with the process terminating thereafter. Otherwise, the process terminates with normal operation of the data processing system.

For example, power to the processing unit could be 30 shut off if the software for processing work units is not detected when the system initializes.

Turning next to **Figure 12**, a flowchart of a process

used for enforcing a policy for processing work units is depicted in accordance with a preferred embodiment of the present invention. The process illustrated in **Figure 12** may be implemented either on a server managing the  
5 distribute computing array, such as server **500** in **Figure 5** or in a data processing system within the distributed computing array, such as data processing system **502** in **Figure 5**.

The process begins by initializing a breach counter  
10 (step **1200**). The breach counter is used to track how often a failure to follow a policy for availability of a system occurs. Next, work units processed over a period of time are identified (step **1202**). A determination is then made as to whether the number of work units  
15 processed is less than a threshold (step **1204**). If the number of work units processed is less than the threshold, then the breach counter is incremented (step **1206**).

Thereafter, a determination is made is to whether  
20 the breach counter is greater than a threshold for a number of allowable breaches (step **1208**). This determination is used to allow a owner of a subsidized computer an opportunity to comply with the policy. If the breach counter is greater than the threshold, then a  
25 disable mechanism is initiated (step **1210**) with the process terminating thereafter.

Turning back to step **1208**, if the breach counter is not greater than the threshold, a warning is presented to the user (step **1212**). This warning may take the form of  
30 a pop-up message or e-mail message that informs the computer owner that the system has not been available to

09352574 09352574

process the required number of work units over a period of time. Thereafter, the process waits for some period of time before proceeding (step **1214**). The process then returns to step **1202** as described above. Turning back to step **1204**, if the number of work units processed is not less than a threshold, the breach counter is set equal to zero (step **1216**) with the process then returning to step **1202**.

Thus, the present invention provides a method, apparatus, and computer implemented instructions for building and providing a distributed computing array. This mechanism provides benefits both to the user and the business creating the distributed computing array. The users are provided with a no cost or low cost system, while the business is relieved of the burden of providing space, power, and maintenance for the computers.

It is important to note that while the present invention has been described in the context of a fully functioning data processing system, those of ordinary skill in the art will appreciate that the processes of the present invention are capable of being distributed in the form of a computer readable medium of instructions and a variety of forms and that the present invention applies equally regardless of the particular type of signal bearing media actually used to carry out the distribution. Examples of computer readable media include recordable-type media, such as a floppy disk, a hard disk drive, a RAM, CD-ROMs, DVD-ROMs, and transmission-type media, such as digital and analog communications links, wired or wireless communications links using transmission forms, such as, for example, radio frequency and light wave transmissions. The

computer readable media may take the form of coded formats that are decoded for actual use in a particular data processing system.

The description of the present invention has been  
5 presented for purposes of illustration and description, and is not intended to be exhaustive or limited to the invention in the form disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art. The embodiment was chosen and described in  
10 order to best explain the principles of the invention, the practical application, and to enable others of ordinary skill in the art to understand the invention for various embodiments with various modifications as are suited to the particular use contemplated.